Con el fin de poner en disponibilidad los resultados del modelo para ser utilizados en páginas web o apps es necesario implementar herramientas de DevOps para facilitar la implementación y la automatización de las tareas. Por otro lado, es necesario crear un pipeline en una plataforma de computación en la nube como AWS; Esto debido a las ventajas de escalabilidad, velocidad, entrega rápida, confiabilidad y bajo costo que ofrece.

Para empezar, se propone el siguiente pipeline de diseño de datos:

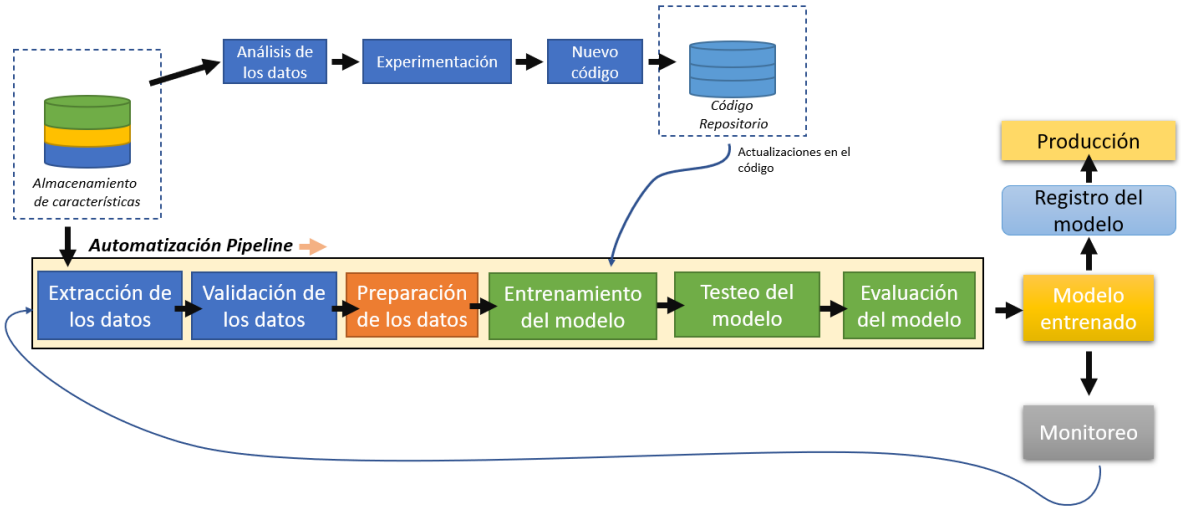

Figura 1. Diagrama de automatización del Pipeline.

En la figura 1 se muestran los pasos a seguir para la automatización del modelo en AWS, donde el pipeline contiene desde la extracción de los datos hasta la evaluación del modelo. Por otro, lado en la parte superior se muestra como es posible realizar actualizaciones y modificaciones en el código para mejorar el modelo desde un repositorio (*Por ejemplo: GitHub*). Adicionalmente, el monitoreo constante es necesario cuando el modelo no esta funcionando de manera adecuada, por lo que es necesario volver a lanzar el pipeline de nuevo con unos nuevos datos.

En AWS esto se implementa realizando lo siguiente, para no depender de la capacidad computacional local. Se crea una instancia EC2 de tipo T2.micro de uso gratuito (dependiendo del fin que se necesite), se escoge por defecto el Key Pair (login) vokey y por último se lanza la instancia. Por otro lado, es necesario crear espacio donde almacenar en la nube que se conoce como S3, de esta manera se le asigna un nombre único; en esta sección se puede cargar las bases de datos o los archivos de Excel que contengan tablas que serán utilizadas para entrenar los modelos. Como tercer punto, se busca Amazon SageMaker para crear una instancia de bloc de notas con el fin de cargar el Jupyter notebook creado específicamente para SageMakerStudio, lo cual implica un cambio en algunas funciones y librerías que se abrían utilizado en Jupyternotebook. Adicionalmente, es posible utilizar las funcionalidades sencillas como Autopilot, el cual genera un notebook con una solución e implementación de un modelo de machine Learning. Finalmente, el despliegue de los modelos se hace por el método deploy, donde se especifica la cantidad de instancias iniciales que se desean tener y además se especifica el tamaño de la instancia (el cual se puede modificar) - $predictor = model.deploy(1,'ml.m4.large')$. En la figura 2, se muestra el proceso descrito anteriormente.

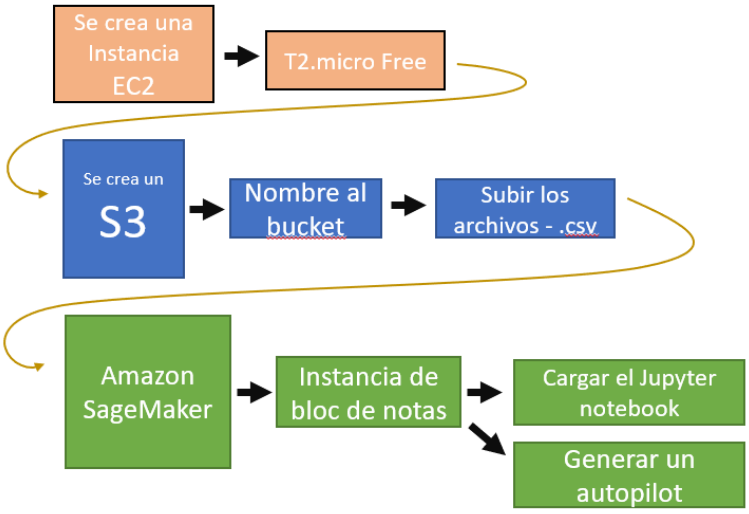

Figura 2. Diagrama implementación Amazon SageMaker.

Productizar

Para poder mandar a producción el modelo de ML, es necesario crear una función lambda para que llame al Endpoint de SageMaker. Primero, se define el lenguaje del código – Python 3.6 y se crea un nuevo role y se ingresan los permisos de políticas para invocar el Endpoint del modelo. En este caso, en la implementación del modelo de machine Learning es necesario generar un código que incluya el Endpoint ; el cual se puede obtener ya sea especificando un endpoint para una API o del servidor de la región de AWS. AWS tiene un ejemplo de cómo obtener esta función de lambda. Después, dentro del apartado de environment variables se ingresa la clave y el valor del endpoint, donde la clave es un nombre único y el valor es el que se obtiene anteriormente.

Finalmente, en el apartado de Amazon API Gateway, se crea una API, donde se le asigna un nombre único. Posteriormente, se crea un nuevo Resource con un nombre para el path y el mismo nombre para el name. A continuación, se crea un método y se la agrega la opción post. Dentro de la configuración de post, es posible seleccionar el tipo de integración- Lambda Funtion y el nombre de mi función lambda. Por utimo, en acciones en el apartado de aplicaciones seleccionamos DeployAPI agregamos- Deployment stage : New Stage –, – Deployment description : Stage Name y seleccionamos Deploy para proporcionar la URL de invocación.

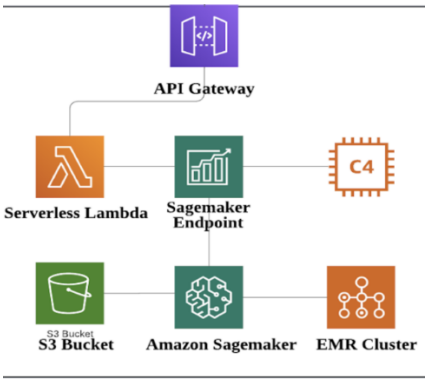

Figura 3. AWS SageMaker, Lambda y API Gateway